



# Demystifying Gluster

## GlusterFS and RHS for the SysAdmin

Niels de Vos

Sr. Software Maintenance Engineer, Red Hat

Gluster Cloud Night Amsterdam – 2014-03-04

# Agenda

- Technology Overview
- Scaling Up and Out
- A Peek at GlusterFS Logic
- Redundancy and Fault Tolerance
- Data Access
- General Administration
- Use Cases
- Common Pitfalls

# Technology Overview

## Demystifying Gluster

GlusterFS and RHS for the SysAdmin

# What is GlusterFS?

- Scalable, general-purpose storage platform
  - POSIX-y Distributed File System
  - Object storage (swift)
  - Distributed block storage (qemu)
  - Flexible storage (libgfapi)
- No Metadata Server
- Heterogeneous Commodity Hardware
- Standards-Based – Clients, Applications, Networks
- Flexible and Agile Scaling
  - Capacity – Petabytes and beyond
  - Performance – Thousands of Clients

# What is Red Hat Storage?

- Enterprise Implementation of GlusterFS
- Software Appliance
- Bare Metal Installation
- Built on RHEL + XFS
- Subscription Model
- Storage Software Appliance
  - Datacenter and Private Cloud Deployments
- Virtual Storage Appliance
  - Amazon Web Services Public Cloud Deployments

# GlusterFS vs. Traditional Solutions

- A basic NAS has limited scalability and redundancy
- Other distributed filesystems limited by metadata
- SAN is costly & complicated but high performance & scalable
- GlusterFS =
  - Linear Scaling
  - Minimal Overhead
  - High Redundancy
  - Simple and Inexpensive Deployment

# Technology Stack

## Demystifying Gluster

GlusterFS and RHS for the SysAdmin

# Terminology

- Brick
  - A filesystem mountpoint
  - A unit of storage used as a GlusterFS building block
- Translator
  - Logic between the bits and the Global Namespace
  - Layered to provide GlusterFS functionality
- Volume
  - Bricks combined and passed through translators
- Node / Peer
  - Server running the gluster daemon and sharing volumes



# Disk, LVM, and Filesystems

- Direct-Attached Storage (DAS)
  - or-
- Just a Bunch Of Disks (JBOD)
- Hardware RAID
  - *RHS: RAID 6 required*
- Logical Volume Management (LVM)
- XFS, EXT3/4, BTRFS
  - Extended attributes support required
  - *RHS: XFS required*

# Gluster Components

- glusterd
  - Elastic volume management daemon
  - Runs on all export servers
  - Interfaced through gluster CLI
- glusterfsd
  - GlusterFS brick daemon
  - One process for each brick
  - Managed by glusterd

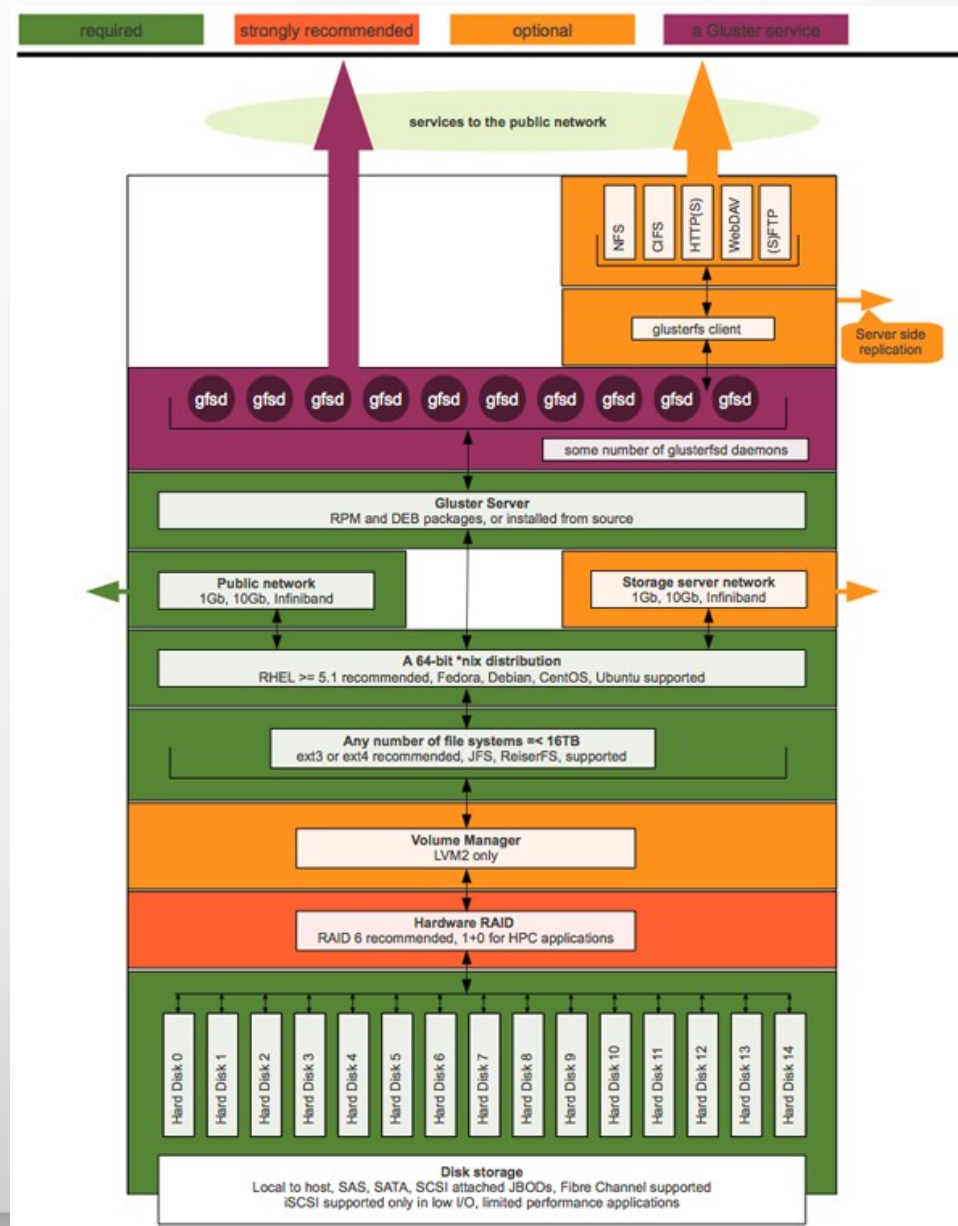
# Gluster Components

- glusterfs
  - NFS server daemon
  - FUSE client daemon
  - Proactive self-heal daemon
- mount.glusterfs
  - FUSE native mount tool
- gluster
  - Gluster Console Manager (CLI)

# Data Access Overview

- GlusterFS Native Client
  - Filesystem in Userspace (FUSE)
- NFS
  - Built-in Service
- SMB/CIFS
  - Samba server required
- Unified File and Object (UFO)
  - Simultaneous object-based access
- NEW! libgfapi flexible abstracted storage
  - Integrated in upstream Samba and Ganesha-NFS

# Putting it All Together



# Scaling Up

- Add disks and filesystems to a node
- Expand a GlusterFS volume by adding bricks

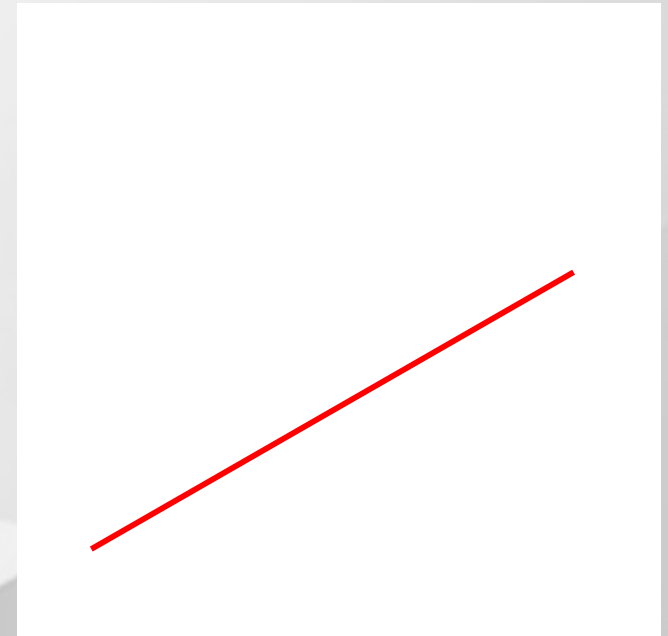
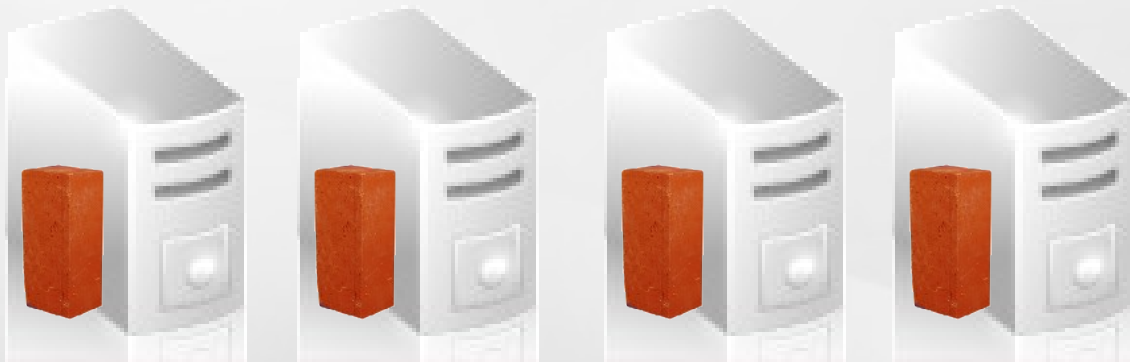


XFS



# Scaling Out

- Add GlusterFS nodes to trusted pool
- Add filesystems as new bricks



# Under the Hood

## Demystifying Gluster

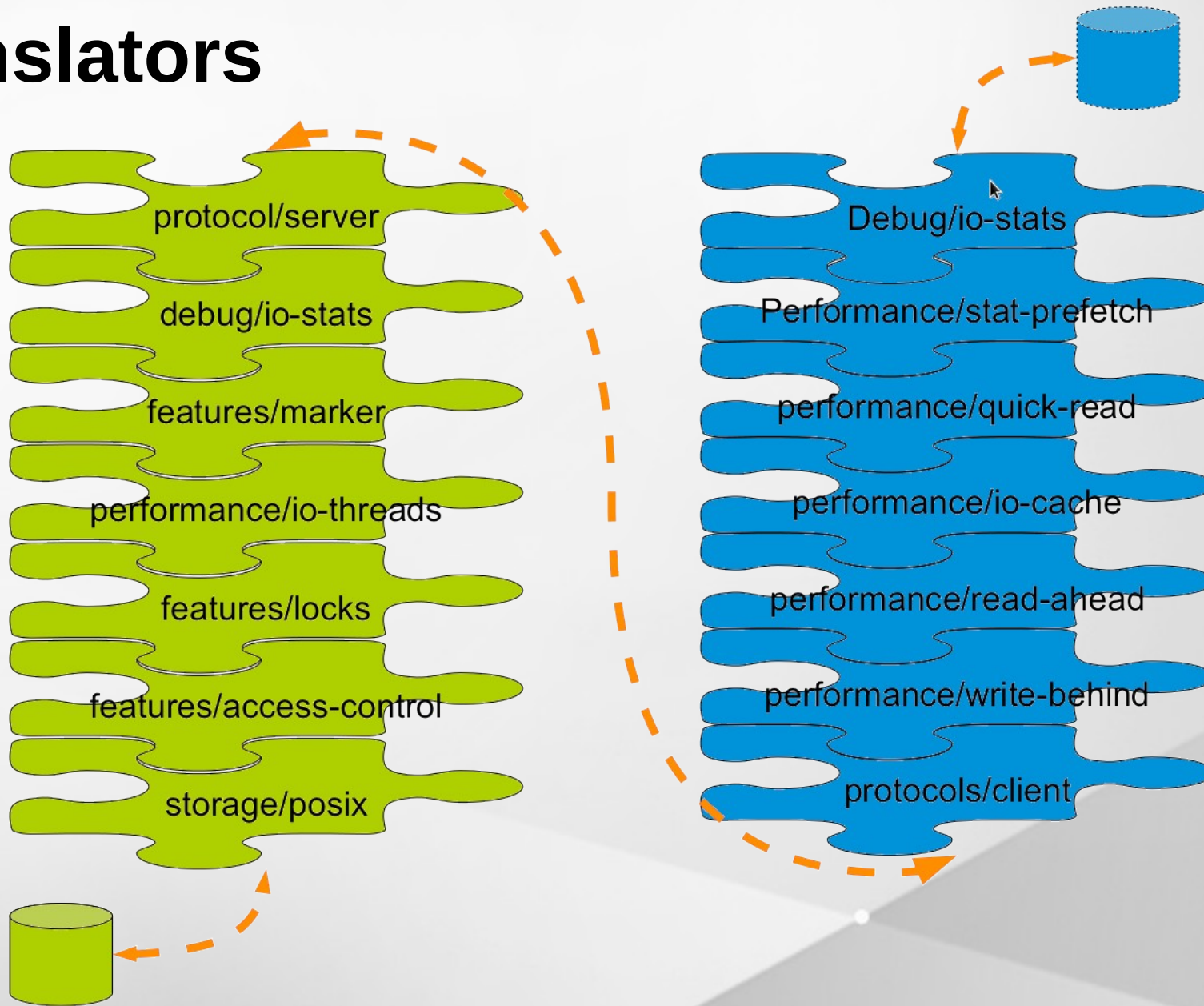
GlusterFS and RHS for the SysAdmin



# Elastic Hash Algorithm

- No central metadata
  - No Performance Bottleneck
  - Eliminates risk scenarios
- Location hashed intelligently on path and filename
  - Unique identifiers, similar to md5sum
- The “Elastic” Part
  - Files assigned to virtual volumes
  - Virtual volumes assigned to multiple bricks
  - Volumes easily reassigned on the fly

# Translators

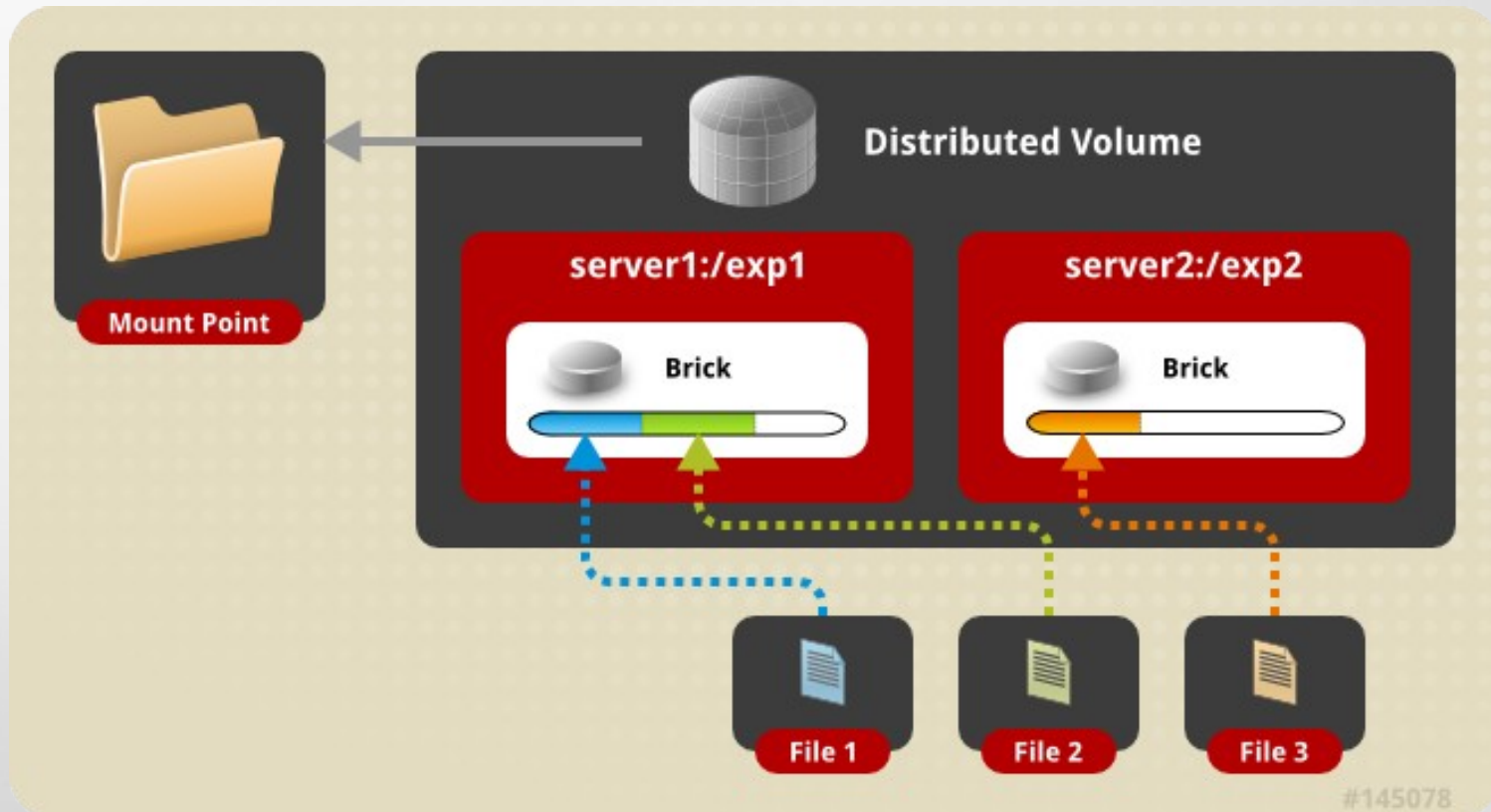


# Basic Volumes

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

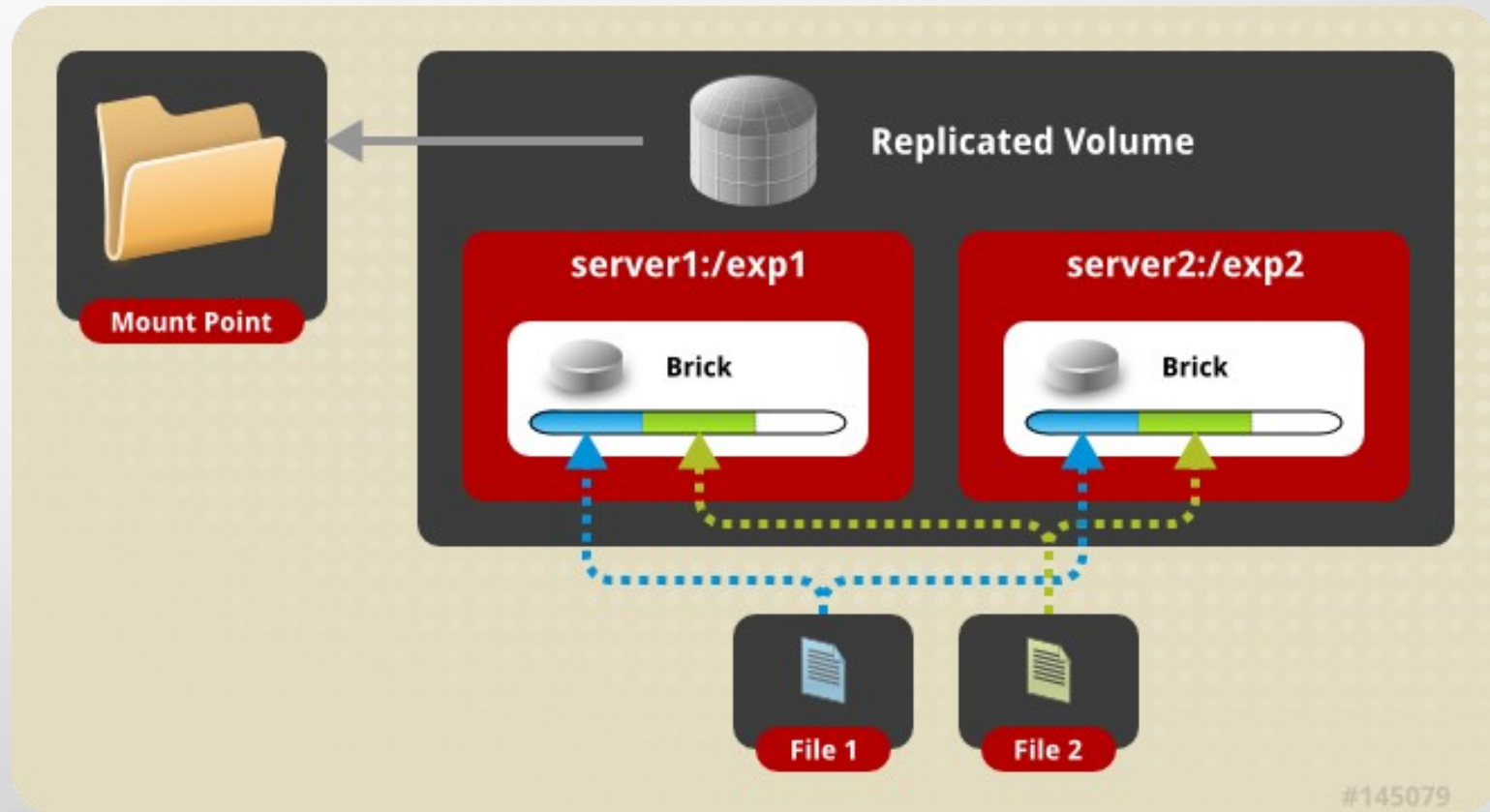
# Distributed Volume

- Files “evenly” spread across bricks
- *Similar* to file-level RAID 0
- Server/Disk failure could be catastrophic



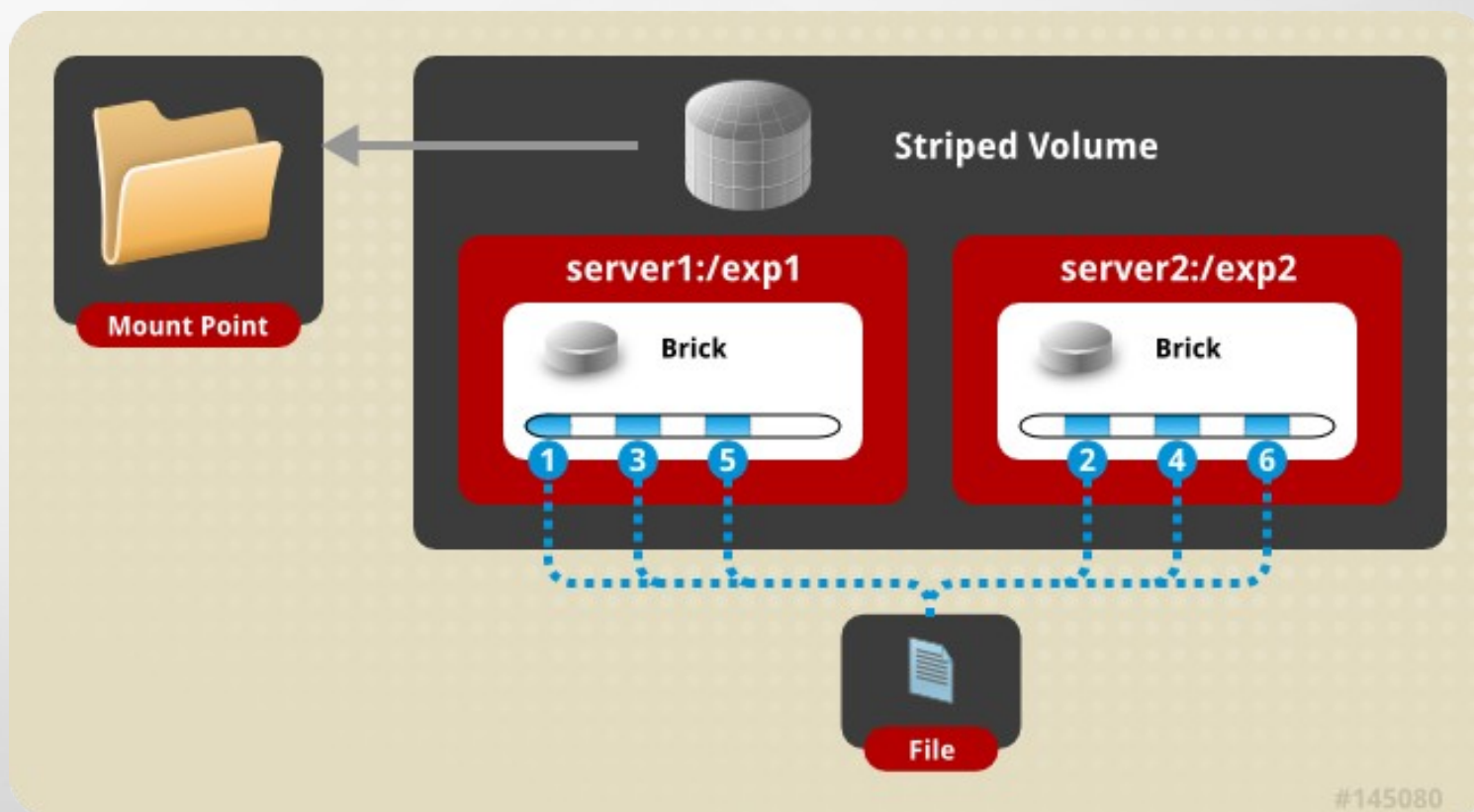
# Replicated Volume

- Copies files to multiple bricks
- *Similar* to file-level RAID 1



# Striped Volumes

- Individual files split among bricks
- *Similar* to block-level RAID 0
- *Limited Use Cases* – HPC Pre/Post Processing



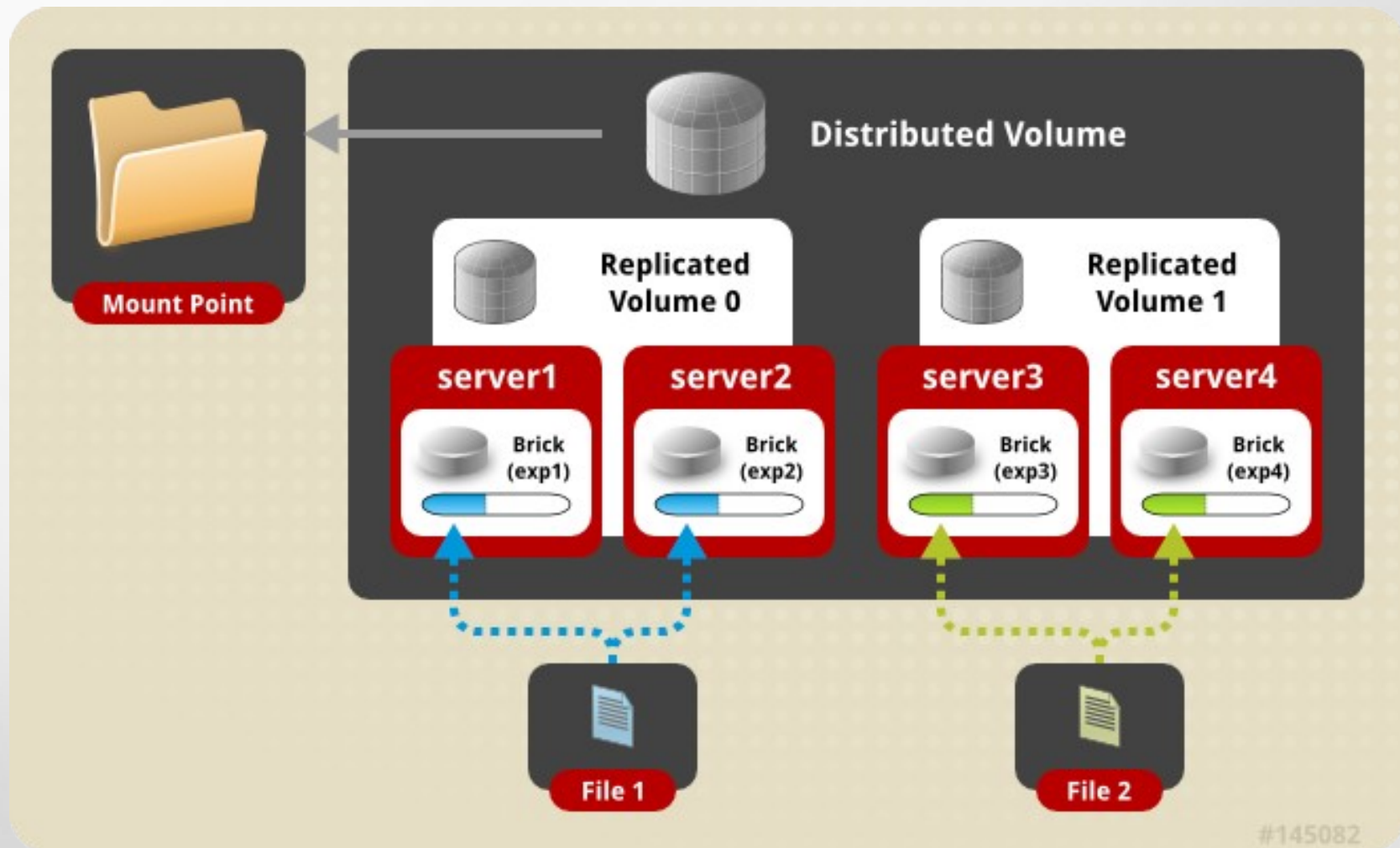


# Layered Functionality

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

# Distributed Replicated Volume

- Distributes files across replicated bricks

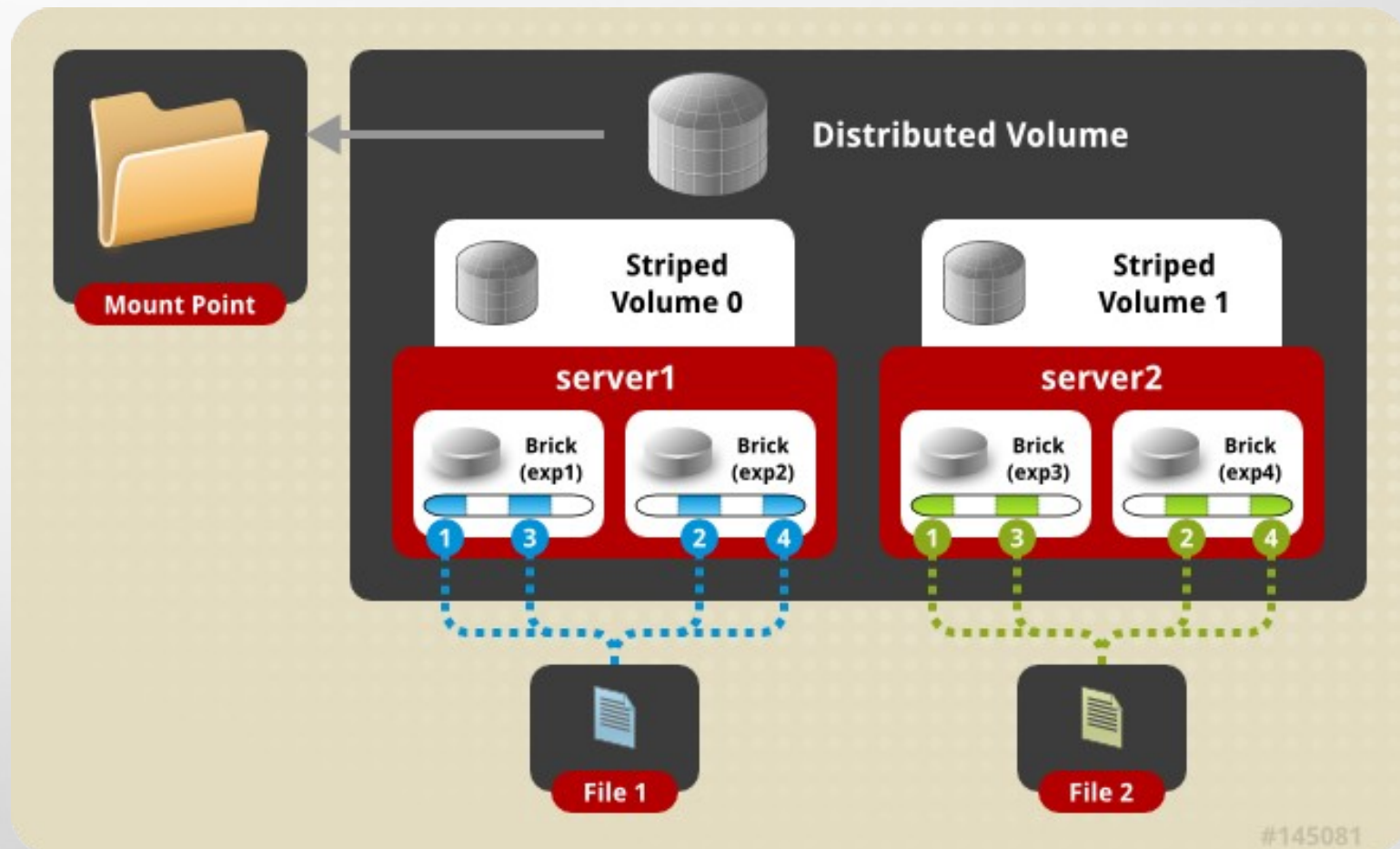


#145082



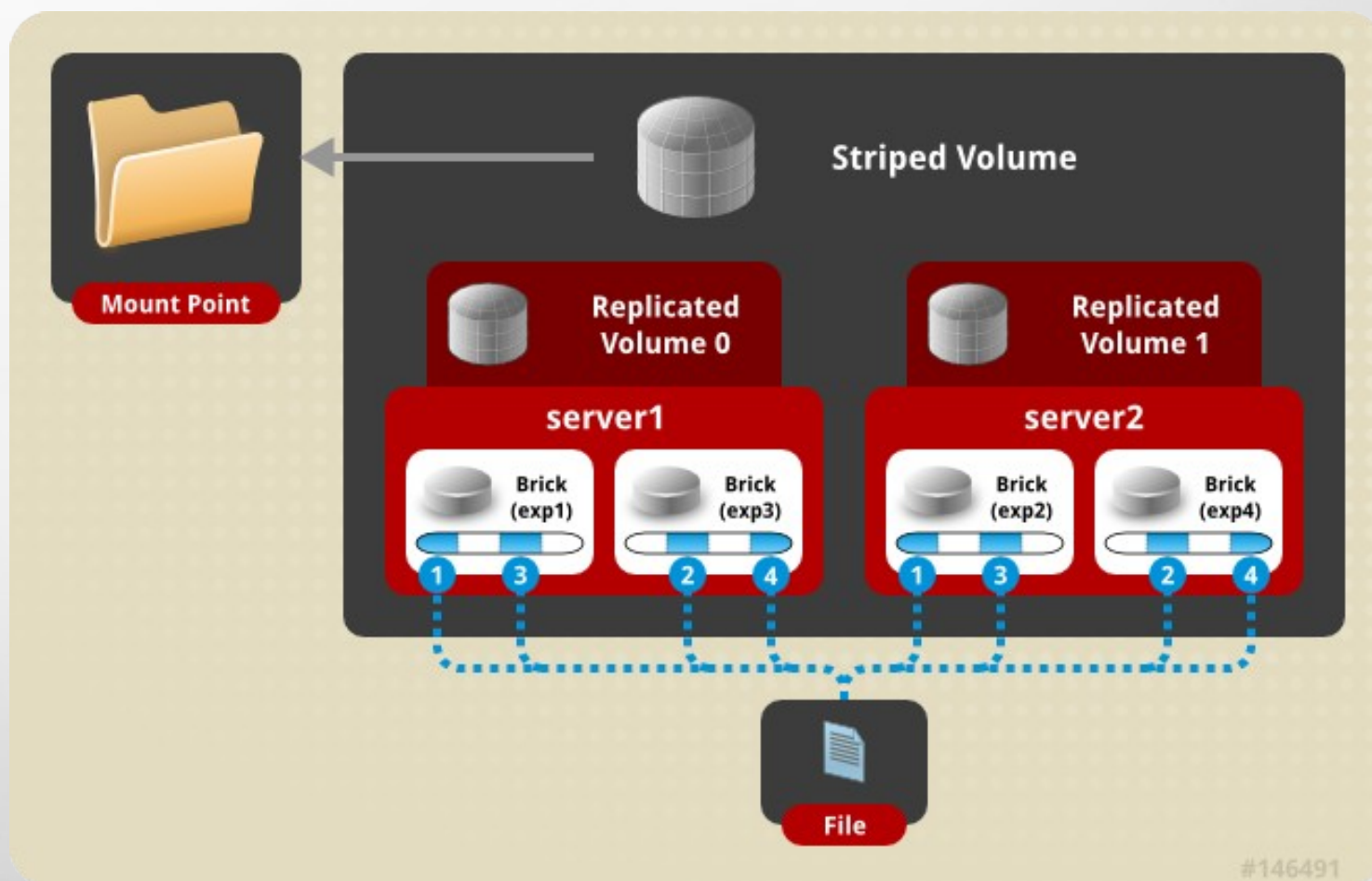
# Distributed Striped Volume

- Files striped across two or more nodes
- Striping plus scalability



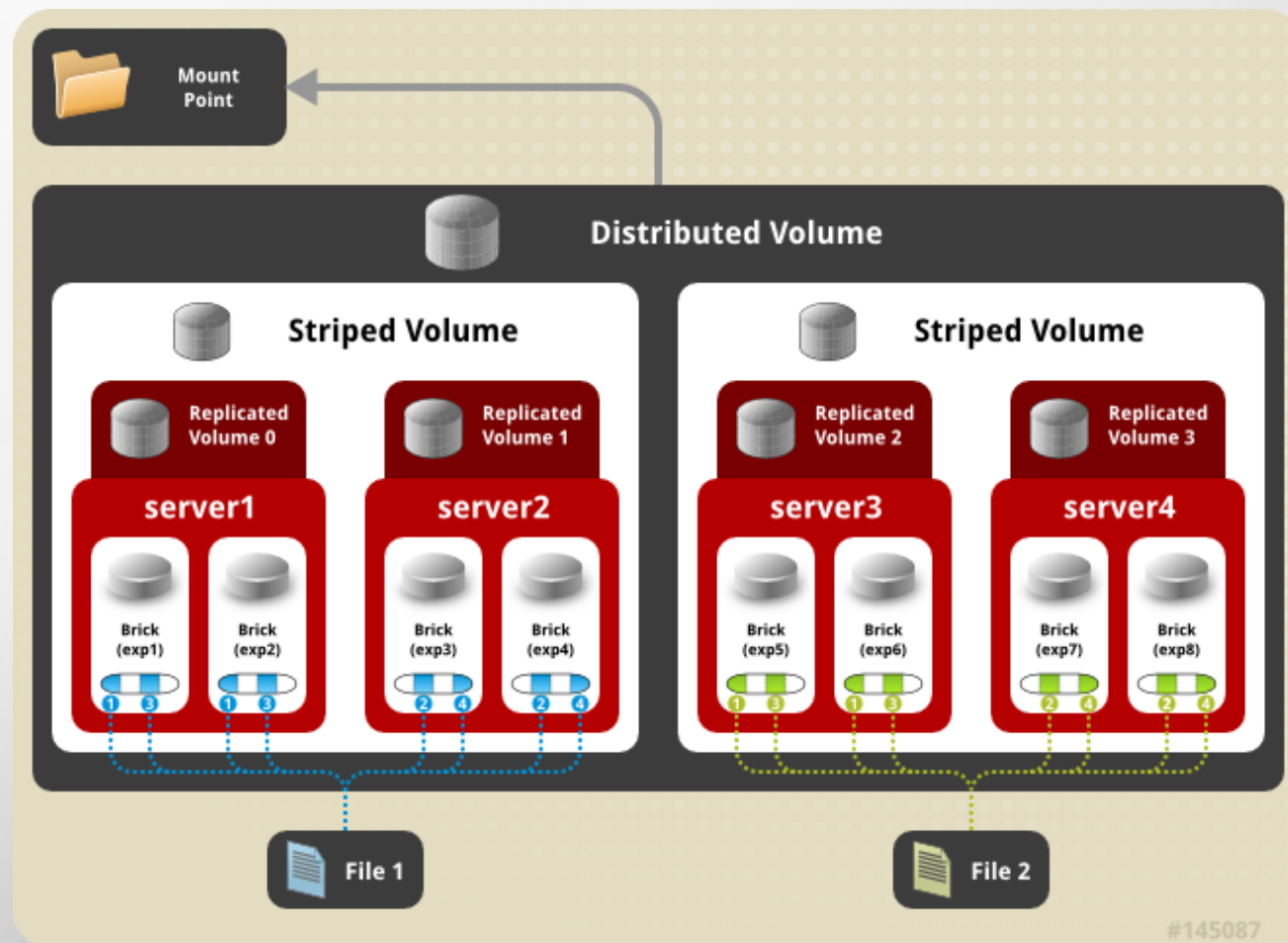
# Striped Replicated Volume

- RHS 2.0 / GlusterFS 3.3+
- *Similar* to RAID 10 (1+0)



# Distributed Striped Replicated Volume

- RHS 2.0 / GlusterFS 3.3+
- *Limited Use Cases* – Map Reduce

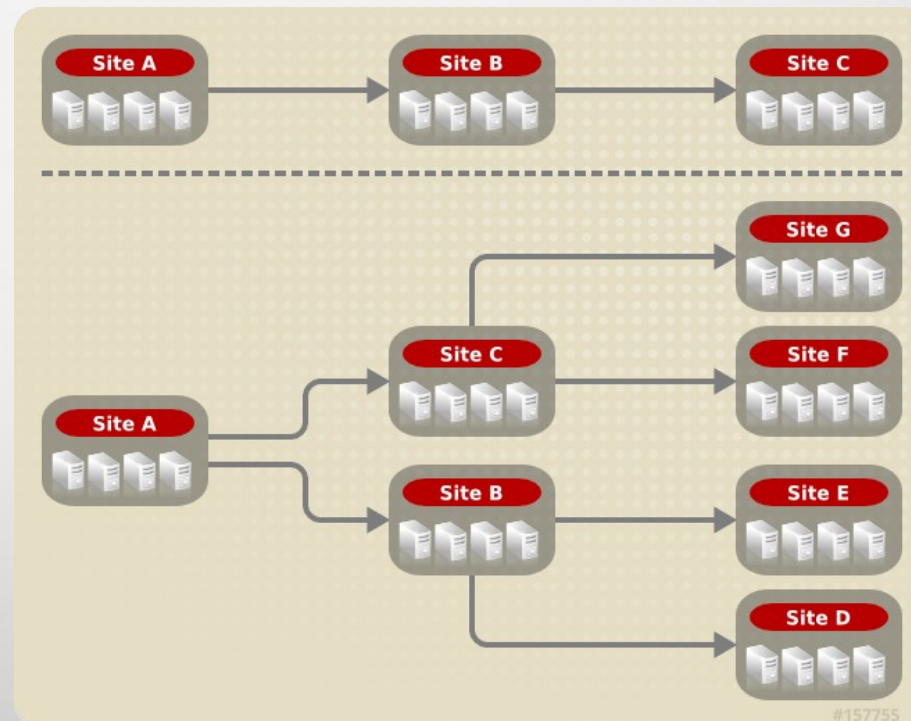


# Asynchronous Offsite for DR and Archive

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

# Geo Replication

- Asynchronous across LAN, WAN, or Internet
- Master-Slave model -- Cascading possible
- Continuous and incremental
- Data is passed between defined master and slave **only**



# Replicated Volumes vs Geo-replication

Replicated Volumes	Geo-replication
Mirrors data across clusters	Mirrors data across geographically distributed clusters
Provides high-availability	Ensures backing up of data for disaster recovery
Synchronous replication (each and every file operation is sent across all the bricks)	Asynchronous replication (checks for the changes in files periodically and syncs them on detecting differences)



# Data Access

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

# GlusterFS Native Client (FUSE)

- FUSE kernel module allows the filesystem to be built and operated entirely in userspace
- Specify mount to any GlusterFS server
- Native Client fetches volume layout (`.vol` file) from the mounted server
- Communicates directly with all bricks to access data
- Recommended for high concurrency and high write performance
- Load is inherently balanced across distributed volumes



# NFS

- Standard NFS v3 clients
  - *Note: Mount with vers=3 option*
- Standard automounter is supported
- Mount to any storage server, or use a load balancer
- GlusterFS NFS server includes Network Lock Manager (NLM) to synchronize locks across clients
- Better performance for reading many small files from a single client
- Load balancing must be managed externally
  - Round Robin DNS and/or virtual IP-address(es)

# NEW! libgfapi

- Introduced with GlusterFS 3.4
- User-space library for accessing data in GlusterFS
- Filesystem-like API
- Runs in application process
- no FUSE, no copies, no context switches
- ...but same volfiles, translators, etc.

# General Administration

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

# Preparing a Brick

```
# lvcreate -L 100G -n lv_brick1 vg_server1
# mkfs -t xfs -i size=512 /dev/vg_server1/lv_brick1
# mkdir /brick1
# mount /dev/vg_server1/lv_brick1 /brick1
# echo '/dev/vg_server1/lv_brick1 /brick1 xfs defaults 1 2' >> /etc/fstab
```

# Adding Nodes (peers) and Volumes

## Peer Probe

```
gluster> peer probe server3
gluster> peer status
Number of Peers: 2

Hostname: server2
Uuid: 5e987bda-16dd-43c2-835b-08b7d55e94e5
State: Peer in Cluster (Connected)

Hostname: server3
Uuid: 1e0ca3aa-9ef7-4f66-8f15-cbc348f29ff7
State: Peer in Cluster (Connected)
```

## Distributed Volume

```
gluster> volume create my-dist-vol server2:/brick2 server3:/brick3
gluster> volume info my-dist-vol
Volume Name: my-dist-vol
Type: Distribute
Status: Created
Number of Bricks: 2
Transport-type: tcp
Bricks:
Brick1: server2:/brick2
Brick2: server3:/brick3
gluster> volume start my-dist-vol
```

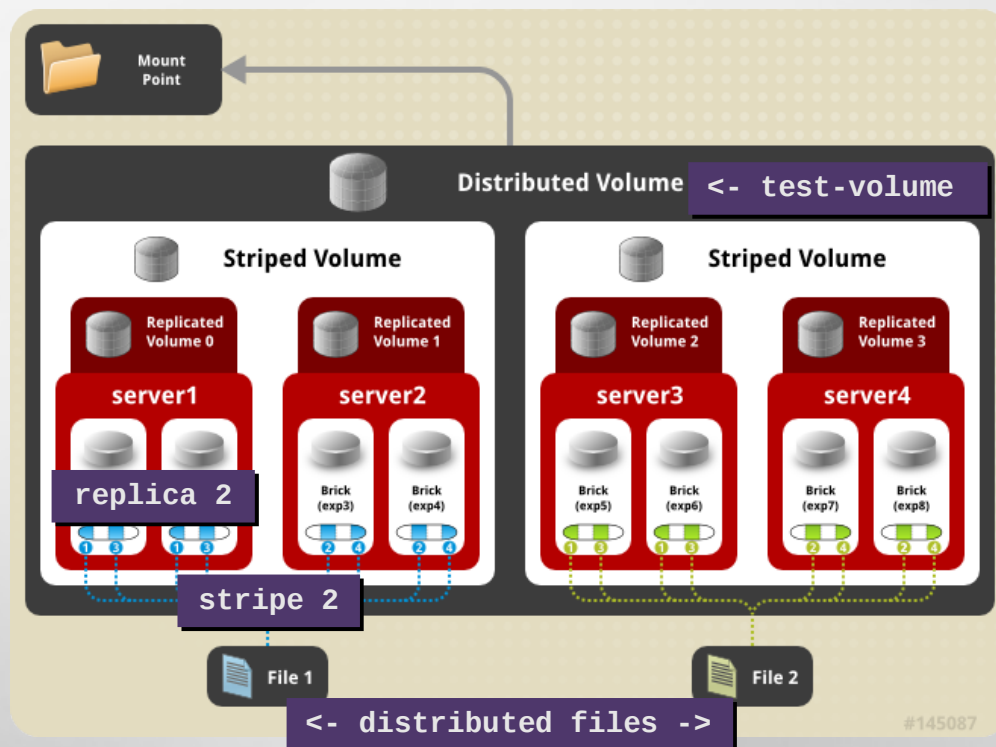
# Distributed Striped Replicated Volume

```
gluster> volume create test-volume replica 2 stripe 2 transport tcp \  
server1:/exp1 server1:/exp2 server2:/exp3 server2:/exp4 \  
server3:/exp5 server3:/exp6 server4:/exp7 server4:/exp8
```

Multiple bricks of a replicate volume are present on the same server. This setup is not optimal.

Do you still want to continue creating the volume? (y/n) y

Creation of volume test-volume has been successful. Please start the volume to access data.



# Distributed Striped Replicated Volume

```
gluster> volume create test-volume stripe 2 replica 2 transport tcp \  
server1:/exp1 server2:/exp3 server1:/exp2 server2:/exp4 \  
server3:/exp5 server4:/exp7 server3:/exp6 server4:/exp8  
Creation of volume test-volume has been successful. Please start the volume to access  
data.
```

```
gluster> volume info test-volume  
  
Volume Name: test-volume  
Type: Distributed-Striped-Replicate  
Volume ID: 8f8b8b59-d1a1-42fe-ae05-abe2537d0e2d  
Status: Created  
Number of Bricks: 2 x 2 x 2 = 8  
Transport-type: tcp  
Bricks:  
Brick1: server1:/exp1  
Brick2: server2:/exp3  
Brick3: server1:/exp2  
Brick4: server2:/exp4  
Brick5: server3:/exp5  
Brick6: server4:/exp7  
Brick7: server3:/exp6  
Brick8: server4:/exp8
```

# Manipulating Bricks in a Volume

```
gluster> volume add-brick my-dist-vol server4:/brick4
```

```
gluster> volume rebalance my-dist-vol fix-layout start
```

```
gluster> volume rebalance my-dist-vol start
```

```
gluster> volume rebalance my-dist-vol status
```

Node	Rebalanced-files	size	scanned	failures	status
localhost	112	15674	170	0	completed
10.16.156.72	140	3423	321	2	completed

```
gluster> volume remove-brick my-dist-vol server2:/brick2 start
```

```
gluster> volume remove-brick my-dist-vol server2:/brick2 status
```

Node	Rebalanced-files	size	scanned	failures	status
localhost	16	16777216	52	0	in progress
192.168.1.1	13	16723211	47	0	in progress

```
gluster> volume remove-brick my-dist-vol server2:/brick2 commit
```



# Migrating Data / Replacing Bricks

```
gluster> volume replace-brick my-dist-vol server3:/brick3 server5:/brick5 start
gluster> volume replace-brick my-dist-vol server3:/brick3 server5:/brick5 status
Current File = /usr/src/linux-headers-2.6.31-14/block/Makefile
Number of files migrated = 10567
Migration complete
gluster> volume replace-brick my-dist-vol server3:/brick3 server5:/brick5 commit
```

# Volume Options

## Auth

```
gluster> volume set my-dist-vol auth.allow 192.168.1.*  
gluster> volume set my-dist-vol auth.reject 10.*
```

## NFS

```
gluster> volume set my-dist-vol nfs.volume-access read-only  
gluster> volume set my-dist-vol nfs.disable on
```

## Other advanced options

```
gluster> volume set my-dist-vol features.read-only on  
gluster> volume set my-dist-vol performance.cache-size 67108864
```

# Volume Top Command

```
gluster> volume top my-dist-vol read brick server3:/brick3 list-cnt 3
Brick:  server:/export/dir1
        =====Read file stats=====

read      filename
call count

116       /clients/client0/~dmtmp/SEED/LARGE.FIL
64        /clients/client0/~dmtmp/SEED/MEDIUM.FIL
54        /clients/client2/~dmtmp/SEED/LARGE.FIL
```

- Many top commands are available for analysis of files, directories, and bricks
- Read and write performance test commands available
  - Perform active dd tests and measure throughput

# Volume Profiling

```
gluster> volume profile my-dist-vol start
```

```
gluster> volume profile my-dist-vol info
```

```
Brick: Test:/export/2
```

```
Cumulative Stats:
```

Block Size:	1b+	32b+	64b+
Read:	0	0	0
Write:	908	28	8

```
...
```

%-latency	Avg-latency	Min-Latency	Max-Latency	calls	Fop
4.82	1132.28	21.00	800970.00	4575	WRITE
5.70	156.47	9.00	665085.00	39163	READDIRP
11.35	315.02	9.00	1433947.00	38698	LOOKUP
11.88	1729.34	21.00	2569638.00	7382	FXATTROP
47.35	104235.02	2485.00	7789367.00	488	FSYNC

```
-----
```

```
Duration      : 335
```

```
BytesRead     : 94505058
```

```
BytesWritten  : 195571980
```

# Use Cases

## Demystifying Gluster

GlusterFS and RHS for the SysAdmin

# Common Solutions

- Media / Content Distribution Network (CDN)
- Backup / Archive / Disaster Recovery (DR)
- Large Scale File Server
- Home directories
- High Performance Computing (HPC)
- Infrastructure as a Service (IaaS) storage layer

# Hadoop – Map Reduce

- Access data within and outside of Hadoop
- No HDFS name node single point of failure / bottleneck
- Seamless replacement for HDFS
- Scales with the massive growth of big data

# Common Pitfalls

**Demystifying Gluster**  
GlusterFS and RHS for the SysAdmin

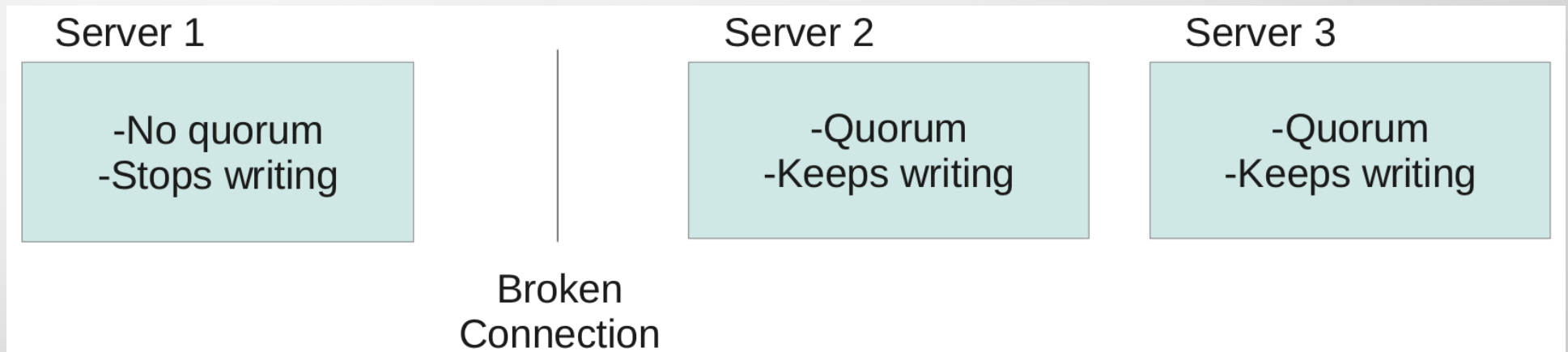


# Split-Brain Syndrome

- Communication lost between replicated peers
- Clients write separately to multiple copies of a file
- No automatic fix
  - May be subjective which copy is right – ALL may be!
  - Admin determines the “bad” copy and removes it
  - Self-heal will correct the volume
    - Trigger a recursive stat to initiate
    - Proactive self-healing in RHS 2.0 / GlusterFS 3.3

# Quorum Enforcement

- Disallows writes (EROFS) on non-quorum peers
- Significantly reduces files affected by split-brain
- Preferred when data integrity is the priority
- Not preferred when application integrity is the priority



# NEW! Server-Side Quorum

- In GlusterFS 3.3
  - Client-side
  - Replica set level
- NOW in GlusterFS 3.4
  - Server-side
  - Cluster-level (glusterd)

# Your Storage Servers are Sacred!

- Don't touch the brick filesystems directly!
- They're Linux servers, but treat them like appliances
  - Separate security protocols
  - Separate access standards
- Don't let your Jr. Linux admins in!
  - A well-meaning sysadmin can quickly break your system or destroy your data

**Do it!**

# **Demystifying Gluster**

## **GlusterFS and RHS for the SysAdmin**



# Do it!

- Build a test environment in VMs in just minutes!
- Get the bits:
  - Fedora 19 has GlusterFS packages natively
  - RHS 2.1 ISO available on Red Hat Portal
  - Go upstream: [www.gluster.org](http://www.gluster.org)



# *Thank You!*

*Slides Available at: <http://people.redhat.com/ndevos/talks/>  
(based on the slide deck from Dustin Black)*

- **ndevos@redhat.com**  
**storage-sales@redhat.com**
- **RHS:**  
**[www.redhat.com/storage/](http://www.redhat.com/storage/)**
- **GlusterFS:**  
**[www.gluster.org](http://www.gluster.org)**
- **TAM:**  
**[access.redhat.com/support/](http://access.redhat.com/support/)**



 **@Glusterorg**

 **@RedHatStorage**



 **Gluster**

 **Red Hat Storage**

## **Demystifying Gluster**

**GlusterFS and RHS for the SysAdmin**